UNIVERSITY OF TECHNOLOGY SYDNEY
Faculty of Engineering and Information Technology

Exploring Centralized and Distributed Constraint Propagation Algorithms for Solving Constraint Satisfaction Problems

by

Shufeng Kong

Dissertation directed by Professor Sanjiang Li

A Thesis Submitted
in Partial Fulfillment of the
Requirements for the Degree

Doctor of Philosophy

Sydney, Australia

2018

Certificate of Authorship/Originality

I certify that the work in this thesis has not been previously submitted for a degree nor has it been submitted as a part of the requirements for other degree except as fully acknowledged within the text.

I also certify that this thesis has been written by me. Any help that I have received in my research and in the preparation of the thesis itself has been fully acknowledged. In addition, I certify that all information sources and literature used are quoted in the thesis.

Signature of student:

Production Note:
Signature removed prior to publication.

Dedication

To my beloved mom, Miaoping Deng, for raising me up and her unwavering support.

Acknowledgements

This thesis would not be possible without the help and contribution of many others. Most significantly, I am much indebted to my advisor, Professor Sanjiang Li, for his guidance and support throughout my PhD study. Sanjiang always paves the way for me and also his high standard and constructive criticisms have made me a better researcher and helped me grow as a person.

I am also grateful to my collaborators, Dr. Zhiguo Long, Dr. Jae Hee Lee and Dr. Michael Sioutis. Without their help, my thesis would not be as good as it is now. Zhiguo cheered me up when I was feeling depressed and tried his best to offer me help when I needed it. I learn a lot from Jae Hee about critical thinking and how to write a good research paper. He never just imposes his opinions on me but always tries to persuade me with right reasons.

I owe a great debt of gratitude to Professor Rina Dechter for her insightful ideas and wise advices when I was a visiting student at UC Irvine. Rina is very knowledgeable and far-sighted. She let me explore and pursue the questions I found of interest, but also provided direction when I needed it. It has been a pleasure and a privilege to work with her. I also want to thank the members of Rina's automated reasoning group, Junkyu Lee, Filjor Broka, Bobak Pezeshki and Qi Lou, for their warmest welcome and support when I stayed at UC Irvine.

Last but not least, I could not have done this without the incredible support of my family, especially, my beloved mom. She always believes in me even when I have doubts about myself.

Shufeng Kong

Sydney, Australia, 2018.

List of Publications

Journal Papers

J-1. **Shufeng Kong**, Sanjiang Li, Yongming Li and Zhiguo Long: On tree-preserving constraints. *Annals of Mathematics and Artificial Intelligence*, 81(3-4): 241-271 (2017)

J-2. **Shufeng Kong**, Sanjiang Li and Michale Sioutis: Exploring directional path-consistency for solving constraint networks. *The Computer Journal*, first on-line: 27 December 2017. doi: 10.1093/comjnl/bxx122

J-3. **Shufeng Kong**, Jae Hee Lee and Sanjiang Li: A new distributed algorithm for efficient generalized arc-consistency propagation. *Autonomous Agent and Multi-Agent Systems*, first online: 10 May 2018. doi: 10.1007/s10458-018-9388-x

Conference Papers

C-1. **Shufeng Kong**, Sanjiang Li, Yongming Li and Zhiguo Long: On tree-preserving constraints. In *Proceeding of the 21st International Conference on Principles and Practice of Constraint Programming* (CP'15), pp. 244-261 (2015)

C-2. **Shufeng Kong**, Jae Hee Lee and Sanjiang Li: A deterministic distributed algorithm for reasoning with connected row-convex constraints. In *Proceeding of the 16th International Conference on Autonoumous Agent and Multiagent Systems* (AAMAS'17), pp. 203-211 (2017)

C-3. **Shufeng Kong**, Jae Hee Lee and Sanjiang Li: Multiagent simple temporal problem: the arc-consistency approach. In *Proceeding of the 32th AAAI Conference on Artificial Intelligence* (AAAI'18), New Orleans, Louisiana, USA, February 2-7, 2018.

Contents

Certificate ii

Dedication iii

Acknowledgments iv

List of Publications v

List of Figures x

List of Tables xiv

Abbreviation xv

Abstract xvi

1 Introduction 1

1.1 Related Works 3

1.1.1 Local Consistency Algorithms for CSPs 3

1.1.2 Tractable Subclasses of CSPs 4

1.1.3 Local Consistency Algorithms for DisCSPs 6

1.2 Thesis Outline and Contributions 7

1.2.1 Tree-Preserving Constraints 7

1.2.2 CSPs Solvable with Directional PC 8

1.2.3 A Distributed Partial PC Algorithm for Solving CRC Constraint Networks 9

1.2.4 A Distributed AC Algorithm for Solving Multiagent STPs . . 10

1.2.5 A New Distributed Generalized AC Algorithm 11

1.3 Preliminaries 11
1.3.1 Constraint Satisfaction Problem 11
1.3.2 Local Consistency and Properties of Constraint Networks . . . 15
1.3.3 Triangulated Constraint Networks 17
1.3.4 Connected Row-Convex Constraint Networks 18
1.3.5 Algebraic Closure Properties of Constraints 20

2 Tree-Preserving Constraints **23**

2.1 Contribution 23
2.2 Introduction and Chapter Outline 23
2.3 Preliminaries 27
2.4 Tree-Preserving Constraints 32
2.5 Partial PC for Tree-Preserving Constraints 38
2.6 The Scene Labelling Problem 47
2.7 Algebraic Closure Properties of Tree-Preserving Constraints 49
2.8 Evaluations 56
2.9 Conclusion 59

3 CSPs Solvable with Directional PC **70**

3.1 Contribution 70
3.2 Introduction and Chapter Outline 71
3.3 Preliminaries 73
3.4 The Strong Directional PC Algorithm 75
3.5 Directional PC and Variable Elimination 76
3.6 Weak VEP Classes and Majority-Closed Classes 83
3.7 The Variable Elimination Algorithm **DPC*** 85

3.8 Evaluations 90

3.9 Conclusion 94

4 A Distributed Partial PC Algorithm for Solving CRC Constraint Networks 95

4.1 Contribution 95

4.2 Introduction and Chapter Outline 95

4.3 Preliminaries 98

4.4 An Efficient Centralized Algorithm for CRC Constraints 98

4.5 An Efficient Distributed Algorithm for CRC Constraints 107

4.6 Evaluations 111

4.6.1 Theoretical Comparisons 111

4.6.2 Experimental Comparisons 113

4.6.3 In-depth Evaluation of DΔCRC 114

4.7 Conclusion 115

5 A Distributed AC Algorithm for Solving Multiagent STPs 116

5.1 Contribution 116

5.2 Introduction and Chapter Outline 116

5.3 Preliminaries 118

5.4 Solving the STP with Arc-Consistency 121

5.4.1 A Centralized AC Algorithm for the STP 125

5.5 Solving the MaSTP with Arc-Consistency 128

5.6 Evaluation 132

5.6.1 ACSTP vs. P^{3}C 134

5.6.2 DisACSTP vs. D△PPC 135

5.7 Conclusion 136

6 A New Distributed Generalized AC Algorithm 137

6.1 Contribution 137

6.2 Introduction and Chapter Outline 137

6.3 Preliminaries 140

6.4 A New Distributed Arc-Consistency Algorithm 141

6.4.1 The Algorithm 144

6.4.2 Handling Termination 149

6.5 Analysis of DisAC3.1 154

6.5.1 Privacy of Individual Agents 154

6.5.2 Termination and Correctness 158

6.5.3 Time and Space Complexities 162

6.6 Non-binary Constraints 164

6.6.1 Generalized Arc-Consistency 164

6.6.2 A Distributed Generalized Arc-Consistency Algorithm 167

6.7 Evaluation 171

6.7.1 DisAC3.1 vs. DisAC9 173

6.7.2 DisGAC3.1 vs. GAC2001/3.1 178

6.8 Conclusion 178

7 Conclusion 183

7.1 Directions for Future Research 185

Bibliography 187

List of Figures

1.1 A CSP example and its constraint graph. 13

1.2 A distributed binary CSP example: $V_i = \{v_1, v_2\}, C_i = \{R_{12}\}$, $V_j = \{v_3, v_4\}, C_j = \{R_{34}\}$, $V_w = \{v_7\}$, $V_p = \{v_6\}$, $V_k = \{v_5\}$, $V_q = \{v_8\}$, $V_l = \{v_9\}$, $C_{ij} = \{R_{23}, R_{24}\}$, $C_{ip} = \{R_{26}\}$, $C_{jk} = \{R_{45}\}$, $C_{wp} = \{R_{76}\}$, $C_{pk} = \{R_{65}\}$, $C_{kl} = \{R_{59}\}$ and $C_{kq} = \{R_{58}\}$. 14

1.3 Path-Consistency [13]. 16

1.4 A graph $G = (V, E)$. 18

1.5 A CRC constraint and a non-CRC constraint. 19

2.1 (a) A tree T. Subgraphs with node sets $\{a, b, c, d\}$, $\{a, b, c\}$, and $\{c, b, d\}$ form, respectively, a subtree, a chain, and a path of T. The degree of node b is 3 and the three branches of b are the subtrees with node sets $\{c\}, \{d\}$ and $\{a, e\}$ respectively. (b) A binary constraint R_{xy} between T_x and T_y, where a dashed arrow from a node u in T_x to a node v in T_y indicates that (u, v) is in R_{xy}. Node c is supported under R_{xy} with image $\{f, g, h\}$, and b is unsupported under R_{xy}. 28

2.2 (a) R_{xy} is a chain- but not path-preserving constraint as the the image of the path $\{c, b, d\}$ of T_x is $\{e, f, g, h\}$, which is not a path of T_y; (b) R_{xy} is a path- but not chain-preserving constraint as the image of the chain $\{a, c\}$ of T_x is $\{e, f, g\}$, which is not a chain of T_y; (c) R_{xy} is a tree-preserving but neither path- nor chain-preserving constraint as the image of the path $\{a, b\}$ of T_x, which is also a chain of T_x, is $\{e, f, g, h\}$ that is neither a path nor a chain of T_y. . . 29

2.3 M_{ab} is a trunk of tree T, i.e., the subtree induced by vertices with red colour. 34

2.4 The flow diagram of proof of Theorem 2.2. 35

2.5 A tree-preserving network $\mathcal{N} = \{R_{12}, R_{13}, R_{23}\}$ over forest domains F_1, F_2, and F_3, where each forest domain consists of three trees which contain only one node. 38

2.6 A triangulated graph G, where $\langle v_1, v_2, v_3, v_4, v_5 \rangle$ is a perfect vertex elimination ordering of G. 39

2.7 Illustration of proof of Lemma 2.8. The figure is taken from [13]. . . . 40

2.8 Illustration of proof of Theorem 2.3. 42

2.9 Possible labelled line configurations of a junction in a picture and their corresponding forest structures. 48

2.10 A scene labelling instance and its corresponding constraint network. . 50

2.11 (a) R is a relation from tree domain T to T, and $R = \{(a, a), (b, a), (c, c)\}$. R is closed under the standard majority operation on T, but it is not tree-preserving w.r.t. T. (b) R is a relation from tree domain T to T, and $R = \{(a, a), (b, b), (c, b), (d, a)\}$. R is tree-preserving w.r.t. T, but it is not closed under the standard majority operation on T. 51

2.12 Illustration of proof of proposition 2.8. Broken arrow lines indicate images of the node and empty circles indicates empty support nodes. 52

2.13 Performance evaluation of different local consistency algorithms for solving tree-preserving constraint networks with different densities. We set $n = 100$, $d = 30$ and $l = 0.5$. 58

2.14 Illustration of proof of Lemma 2.7. 63

2.15 Possible configurations of two connected trunks $M_{a,b}$ and $M_{c,d}$, where $M_{a,b} \cup M_{c,d}$ is the whole tree in (c) and (d), a trunk in (a), (b) and (f), and a branch in (e). 66

2.16 Illustration of proof of Lemma 2.13. 67

3.1 A graph coloring problem with domain $D_i = \{red, blue\}$ for $i = 1, 2, 3, 4$ [37]. 76

3.2 Two binary constraint networks $\mathcal{N}$ and $\mathcal{N}_{-4}$. 78

3.3 A constraint graph that is a chain. 78

3.4 An illustration of Example 3.3. 80

3.5 Illustration of proof of Theorem 3.2. 84

3.6 A constraint network $\mathcal{N}$ and its elimination $\mathcal{N}_{-w}$. 88

3.7 Performance comparisons among **DPC***, **SAC3-SDS** and **PC2001** for solving tree-preserving constraint networks. 91

3.8 Performance comparisons among **DPC***, **SAC3-SDS** and **PC2001** for solving random majority-closed constraint networks. 92

4.1 A constraint graph of Example 4.1. 96

4.2 Illustration of proof of Theorem 4.1. 104

4.3 Execution of DΔCRC on Example 4.1. 107

4.4 Performance comparisons between DΔCRC and D-CRC. 112

4.5 Evaluation of our algorithm DΔCRC for different number of agents and network density. We set $n = 30$ and $d = 20$. 114

5.1 An illustration of Example 5.1. Alice, Bob, company X, company Y are four agents, each owning a local simple temporal network. The circles represent variables and edges constraints. Red edges represent constraints that are shared by two different agents. 117

5.2 Evaluation of ACSTP and $\mathsf{P}^3\mathsf{C}$. The y-axes (on the log scale) represent the number constraint checks. 133

5.3 Evaluation of DisACSTP and D△PPC. The y-axis (on the log scale) represent the number of NCCCs. 134

6.1 The standard agent communication graph of the distributed binary CSP in Figure 1.2. 143

6.2 A T-tree for the standard agent communication graph in Figure 6.1. 144

6.3 The relevant parts of variable v_2 of the distributed binary CSP in Figure 1.2. 147

6.4 Illustrations of messages flow. 150

6.5 Time lines of sending and receiving messages, where circles stand for the time points that agent i sends “domain update” messages to agent j, squares stand for the time points that agent j reports the latest received time stamps of “domain update” messages to the root agent via “up to date” messages, and triangles stand for the time points that the root agent receives “up to date” messages from agent j. 152

6.6 A ternary constraint (s, R), where $s = (v_1, v_2, v_3)$ and $R = \{(a, c, e), (a, c, f), (b, d, f)\}$. 165

6.7 Performance comparisons between DisAC3.1 and DisAC9 on random instances. 172

6.8 Number of messages sent by agents w.r.t. height of spanning trees. We set $n = 10, d = 10, \rho = 1.0, \beta = 0.5$ and #agent = 10. 176

6.9 Speed-up vs. number of agents on benchmark problems. 177

List of Tables

3.1 Comparison of time complexities among state-of-the-art algorithms for solving majority-closed constraint networks. 90

6.1 Comparison of time and space complexities between DisAC3.1 and DisAC9. 173

6.2 Performance comparisons between DisAC3.1 and DisAC9 on the DOMINO problem. Note that m is the number of constraints and p is the number of agents used by both the algorithms. 174

6.3 Performance comparisons between DisAC3.1 and DisAC9 on the Radio Link Frequency Assignment Problem. Note that m is the number of constraints and p is the number of agents used by both the algorithms. 174

Abbreviation

(Dis)CSP - (Distributed) Constraint Satisfaction Problem

BCN - Binary Constraint Network

CRC - Connected Row-Convex

AC - Arc-Consistency

PC - Path-Consistency

DPC - Directional Path-Consistency

PPC - Partial Path-Consistency

STN - Simple Temporal Network

MaSTN - Multiagent Simple Temporal Network

ABSTRACT

Constraint propagation is central to the process of solving a constraint satisfaction problem (CSP). It can be used to solve several large tractable classes of CSPs directly and is also predominantly used to reduce the space of combinations that will be explored by a search algorithm. Constraint propagation, also known as local consistency enforcing, is the process of reducing domains of variables, strengthening constraints, or creating new ones. Arc-consistency (AC) and path-consistency (PC) are two well-known forms of local consistency. Designing efficient local consistency algorithms is a central research task in constraint processing. A related important question is to finding large tractable classes that can be solved by enforcing local consistency.

The class of connected row-convex (CRC) constraints defined over linear domains is a prominent tractable class which is a subclass of the class of row-convex constraints. While the class of row-convex constraints is intractable, it was shown that enforcing PC solves the CSPs over CRC constraints. The CRC constraint class is very expressive and can model problems in domains such as temporal reasoning, VLSI design, geometric reasoning, scene labelling as well as logical filtering.

In Chapter 2 we generalize the class of CRC constraints from linear domains to tree domains and obtain the new tractable class of tree-preserving constraints. We show that enforcing PC can transform a consistent tree-preserving constraint network into an equivalent globally consistent network. We also observe that CRC and tree-preserving constraint networks also can be solved by enforcing directional PC (DPC), a weaker form of PC which can be enforced more efficiently. A natural research question then is to characterize CSPs that are solvable with DPC. In Chapter 3 we provide such a characterization and prove that any class of majority-closed constraints is solvable with DPC and thus give a more efficient algorithm for solving these constraints.

In above, we assume that the knowledge about a CSP (i.e. domains and con-

straints) is known by one central agent, which is often not available when the knowledge about the problem is distributed among autonomous agents. Because of privacy reasons, simply collecting all such knowledge from the individual agents is undesirable or impossible. To address the issue, we need to develop distributed algorithms for solving distributed CSPs. We propose in Chapter 4 the first deterministic distributed algorithm to solve multiagent CRC constraint networks. Our algorithm is a distributed partial PC algorithm which can efficiently transform a CRC constraint network into an equivalent constraint network such that all constraints are minimal (i.e., they are the tightest constraints) and all solutions can be generated in a backtrack-free manner.

We then consider the class of simple temporal constraints in Chapter 5, which is closely related to the class of CRC constraints and is widely used in temporal planning and scheduling. In fact, discretized simple temporal constraints over finite domains are CRC constraints. Previous approaches focus on enforcing partial PC or directional PC to solve a simple temporal network (STN). We show that enforcing AC is sufficient to solve an STN, which not only provides a more efficient algorithm for STNs but also provides the first privacy-preserving distributed algorithm for solving multiagent STNs.

While the above algorithms are complete for certain tractable constraint classes, in Chapter 6 we propose a new distributed AC algorithm for general distributed CSPs, which is more efficient and leaks less private information of agents than existing ones. In particular, our new distributed AC algorithm uses a novel termination determination mechanism, which allows the agents to share domains, constraints and communication addresses only with relevant agents. We further extend it to the first distributed algorithm that enforces generalized AC (GAC) on k-ary ($k \geq 2$) distributed CSPs.